NumPy Coding Questions with Solutions

1. Write a NumPy program to create a 5×5 matrix with values 1,2,3,…,25.

import numpy as np matrix = np.arange(1, 26).reshape(5,5) print(matrix)

2. Generate a 4×4 identity matrix using NumPy.

identity_matrix = np.eye(4) print(identity_matrix)

3. Create a 1D array of numbers from 100 to 200 with step size 10.

arr = np.arange(100, 201, 10) print(arr)

4. Generate a random 3×3 matrix and find its determinant.

mat = np.random.rand(3,3) det = np.linalg.det(mat) print(mat) print('Determinant:', det)

5. Create a NumPy array of 10 random integers between 1 and 100.

rand_arr = np.random.randint(1, 101, 10) print(rand_arr)

6. Write a program to reshape a 1D array of size 12 into a 3×4 matrix.

arr = np.arange(12) reshaped = arr.reshape(3,4) print(reshaped)

7. Create two 3×3 matrices and perform matrix multiplication.

a = np.random.randint(1,10,(3,3)) b = np.random.randint(1,10,(3,3)) product = np.dot(a, b) print(a) print(b) print(product)

8. Find eigenvalues and eigenvectors of a given 2×2 matrix using NumPy.

mat = np.array([[4,2],[1,3]]) eigvals, eigvecs = np.linalg.eig(mat) print(eigvals) print(eigvecs)

9. Create a 5×5 matrix with random values and extract its diagonal elements.

mat = np.random.rand(5,5) diagonal = np.diag(mat) print(diagonal)

10. Generate a 1D array and normalize it (scale values between 0 and 1).

arr = np.random.randint(1,100,10) normalized = (arr - arr.min()) / (arr.max() - arr.min()) print(normalized)

11. Write a program to sort a NumPy array by row and column.

mat = np.random.randint(1,100,(3,3)) row_sorted = np.sort(mat, axis=1) col_sorted = np.sort(mat, axis=0) print(row_sorted) print(col_sorted)

12. Find the indices of maximum and minimum values in a NumPy array.

arr = np.random.randint(1,100,10) print(arr) print('Max index:', arr.argmax()) print('Min index:', arr.argmin())

13. Create a 2D array and flatten it using ravel() and flatten().

mat = np.arange(12).reshape(3,4) print(mat.ravel()) print(mat.flatten())

14. Create a 3×3 matrix and compute its inverse using NumPy.

mat = np.random.randint(1,10,(3,3)) inverse = np.linalg.inv(mat) print(inverse)

15. Write a NumPy program to generate a random permutation of numbers 1 to 10.

perm = np.random.permutation(np.arange(1,11)) print(perm)

16. Create a NumPy array with values from 0 to 20 and replace all even numbers with -1.

arr = np.arange(21) arr[arr%2==0] = -1 print(arr)

17. Write a program to compute the dot product of two arrays.

a = np.array([1,2,3]) b = np.array([4,5,6]) dot = np.dot(a,b) print(dot)

18. Generate a 5×5 random matrix and compute its trace.

mat = np.random.randint(1,10,(5,5)) trace = np.trace(mat) print(trace)

19. Write a NumPy program to split a 1D array into 3 equal parts.

arr = np.arange(9) split = np.split(arr,3) print(split)

20. Create a 3D array of shape (3,3,3) and find its mean across axis=0.

arr = np.random.randint(1,10,(3,3,3)) mean = arr.mean(axis=0) print(mean)

21. Write a program to find the cumulative sum of a NumPy array.

arr = np.array([1,2,3,4]) cumsum = arr.cumsum() print(cumsum)

22. Create a 4×4 matrix with random integers and extract its upper triangular matrix.

mat = np.random.randint(1,10,(4,4)) upper = np.triu(mat) print(upper)

23. Write a program to generate a matrix of size 6×6 with checkerboard pattern (0,1).

checker = np.indices((6,6)).sum(axis=0)%2 print(checker)

24. Generate a 3×3 random matrix and apply element-wise square root.

mat = np.random.randint(1,10,(3,3)) sqrt = np.sqrt(mat) print(sqrt)

25. Create a 1D array of 20 elements and reverse it without using Python slicing.

arr = np.arange(20) reversed_arr = np.flip(arr) print(reversed_arr)

26. Write a program to merge two NumPy arrays vertically and horizontally.

a = np.array([[1,2],[3,4]]) b = np.array([[5,6],[7,8]]) vert = np.vstack((a,b)) horiz = np.hstack((a,b)) print(vert) print(horiz)

27. Create a 2D NumPy array and compute row-wise and column-wise sum.

mat = np.array([[1,2,3],[4,5,6],[7,8,9]]) row_sum = mat.sum(axis=1) col_sum = mat.sum(axis=0) print(row_sum) print(col_sum)

28. Write a NumPy program to replace NaN values in an array with the mean of that column.

arr = np.array([[1,np.nan,3],[4,5,np.nan]]) col_mean = np.nanmean(arr,axis=0) inds = np.where(np.isnan(arr)) arr[inds] = np.take(col_mean, inds[1]) print(arr)

29. Generate two 1D arrays and compute cosine similarity using NumPy.

a = np.array([1,2,3]) b = np.array([4,5,6]) cos_sim = np.dot(a,b)/(np.linalg.norm(a)*np.linalg.norm(b)) print(cos_sim)

30. Create a 4×4 array and rotate it by 90 degrees using NumPy.

mat = np.arange(16).reshape(4,4) rotated = np.rot90(mat) print(rotated)

31. Write a program to create a structured NumPy array with fields (name, age, marks).

dt = np.dtype([('name','U10'),('age','i4'),('marks','f4')]) data = np.array([('Alice',20,85.5),('Bob',22,90.0)],dtype=dt) print(data)

32. Generate a random 3×3 matrix and compute its rank.

mat = np.random.randint(1,10,(3,3)) rank = np.linalg.matrix_rank(mat) print(rank)

33. Create a 5×5 random matrix and normalize each row to have unit length.

mat = np.random.rand(5,5) norm = mat/np.linalg.norm(mat,axis=1,keepdims=True) print(norm)

34. Write a NumPy program to check whether two arrays are equal element-wise.

a = np.array([1,2,3]) b = np.array([1,2,4]) print(np.array_equal(a,b))

35. Generate a random dataset of 1000 numbers and compute histogram using NumPy.

data = np.random.randn(1000) hist, bins = np.histogram(data, bins=10) print(hist) print(bins)

36. Create a 2D array and apply broadcasting to add a 1D array.

mat = np.ones((3,3)) arr = np.array([1,2,3]) result = mat + arr print(result)

37. Write a program to find unique values and their counts in a NumPy array.

arr = np.array([1,2,2,3,4,4,4,5]) unique, counts = np.unique(arr, return_counts=True) print(unique) print(counts)

38. Create two arrays and find Pearson correlation coefficient using NumPy.

a = np.array([1,2,3,4,5]) b = np.array([5,4,3,2,1]) correlation = np.corrcoef(a,b)[0,1] print(correlation)

39. Write a NumPy program to compute numerical gradient of a 1D array.

arr = np.array([1,2,4,7,11]) grad = np.gradient(arr) print(grad)

40. Generate a random 3×3 matrix and perform Singular Value Decomposition (SVD).

mat = np.random.randint(1,10,(3,3)) U, S, Vt = np.linalg.svd(mat) print(U) print(S) print(Vt)